|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Assessment Type** | **Date** | **Max. Marks** | **Weightage** | **Remarks** | **CO** |
| Review I | Immediately after CAT 1 | 20 | 20 | Presentation (online) and the slides should be uploaded in moodle(as pdf) for  documentation; the reviews will  also be recorded - not **more than 3**  students per group. | CO2, CO3, CO5 and CO7 |
| Review II | Immediately after CAT 2 | 30 | 30 |
| Review III | Before FAT Lab Examinations | 50 | 50 |

**Project Component**

[Students are advised to start working on the J component. Don’t wait for review dates. Review dates will be notified,]

Workflow

1. Identification of the problem and discussion with the faculty to finalize the problem.

2. Search for existing solutions if any and

3. Propose a novel and innovative solution/s for the problem finalized by overcoming gaps in existing solutions.

4. Develop the software/app using any of the programming language you are comfortable with.

5. Choose few test cases and simulate the developed tool.

6. Record the results and document it.

|  |  |  |  |
| --- | --- | --- | --- |
| **Rubrics** | | | |
| **Review No** | **S. No.** | **Expected**  **Deliverable** | **Max.**  **Marks** |
| 1 | 1. | Problem Identification and Description | 7 |
| 2. | Proposal of an innovative feasible solution(Idea) with a detailed flow diagram | 8 |
| 3. | Organisation of the presentation – reflection of originality | 3 |
| 4. | Timely submission (ppt in pdf format altealst 3 days before to presentation date) | 2 |
| Total Marks | | 20 |
| 2. | 1. | Development of atleast two modules – User Interface Design/Usability etc. | 5 |
| 2. | 50% Code developed in a convenient programming language | 15 |
| 3. | Organization of the presentation – reflection of originality | 5 |
| 4. | Timely submission (ppt in pdf format altealst 3 days before to presentation date) | 5 |
|  | Total Marks | 30 |
| 3. | 1. | Development of all the modules – User Interface Design/Usability etc. | 7 |
| 2. | 100% Code developed in a convenient programming language | 13 |
| 3. | Organization of the presentation – reflection of originality | 5 |
| 4. | Organization and presentation of the final report | 5 |
| 5. | Viva during presentation (individual assessment) | 15 |
| 6. | Timely submission (ppt in pdf format altealst 3 days before to presentation date) | 5 |
|  | Total Marks | | 50 |
|  | **Note:** There may be a slight variation in the mark splitup from time to time as decided by the faculty. | |  |

**Probable Project on CSE2005 Operating System**

Create a linux from scratch and build at least 60% of the system

<http://www.linuxfromscratch.org/hlfs/view/development/index.html>

Add 64-bit support for virtual memory to XV6 OS, which is currently 32-bit

Add journaling support to the XV6 or PintOS

Create an NFS file server

Create a Bash like shell with all functionalities

Build a driver for the audio – look at OSS or ALSA for help

(<http://freecode.com/projects/alsadriver>)

Build a real time kernel with real time kernel with some of these functionalities: interrupt handling, exception handling, scheduler choice, thread support etc.

(<http://ecos.sourceware.org/about.html>)

Investigate the implementation of timing throughout a distributed system and the various protocols available to implement synchronizing. This could include researching the timing behavior of commonly used programming languages and system

An exokernel eliminates the notion that an operating system should provide abstractions on which applications are built. Instead, it concentrates solely on securely multiplexing the raw hardware. Write a term paper on an exokernel and the suitability of exokernels

Helios is a micro-kernel operating system for embedded and multiprocessor systems. Explore the system and add your own useful module to it.

Explore possibilities of load balancing aspect of distributed systems. Heterogeneous networks involve processors with different computing power, different memory capacity, or they have different communication links. How do they diffuse and balance the load?

Allow processes to migrate from one machine to another over the network. You'll need to do something about the various pieces of a process's state, such as file descriptors in xv6.

Implement paging to disk in xv6 , so that processes can be bigger than RAM. Extend your pager with swapping.

Modify xv6 to have kernel-supported threads inside processes

Explore the scheduler’s priority and CPU placement algorithms in Windows.

Simulation of the Storage Virtualization

Remote Syscall Interfacing with Sockets

User interface using Arduino

Linux From Scratch (LFS)

Creating a Linux driver (Starting with audio driver ALSA)

Create a Bash like shell with functionalities

Building a Mini OS

Creating a Shell in C

Analysing the parallel computing Algorithms

Implementing a functionality in OS

Implementing System CallS in Xv6

Building a Mini OS from Scratch

Docker: OS level Virtualization Framework

NFS

USB Driver detection and interface with Bash

Memory Virtualization

Analysing the performance of page replacement algorithms

Nand2Tetris OS building

Userspace Deduplication File System

Understanding Vulnerabilities of an OS under Rootkit Attack